LJUNGBERG & KÖGEL AB

BmCom – User's manual.

Rev. 02

1 Contents

2 Document history 2
3 Overview 3
3.1 Installing the software 3
4 Settings 4
4.1 Communication 4
4.2 Compatibility 4
5 Data Collection 6
6 Configuration 7
6.1 Configuration Tool 7
6.2 Barcode questions 8
6.3 Estimated blood volume 9
7 Firmware Update 10
8 Appendix 1 11
8.1 ISBT-128 standard 11
8.1.1 Some areas of usage 11
8.1.2 Data identifiers 11
8.1.3 Donation Identification Number 12
8.1.4 Manufacturer's Identity and Container Information 12
8.1.5 Lot number 12
8.2 Configuration 13
8.2.1 Example #1 (without ISBT-128 extensions) 13
8.2.2 Example #2 (with ISBT-128 extensions) 15
8.2.3 Example #3 (EBV) 17
8.3 Analyzing collected data 19

2 Document history

Revision History
2012-08-01 AR Rev. 01: First Revision
2012-10-26 DC Rev. 02: Added chapter about new lines. Some minor layout changes.

3 Overview

BmCom is a Windows software for BioMixers 330 and 330-1. It is used to configure the procedure of barcode scanning during blood donation; it performs firmware update, and is also acting as a server, collecting donation through SD-card, LAN-bus, or wireless communication.

The software replaces the older tools BM330inf and BM330conf.

3.1 Installing the software

BmCom requires a PC with:

- Windows XP, Windows Vista or Windows7, 32 or 64bit.
- .Net framework 4.0 Client profile - available through Microsoft Windows Update.

The Setup.exe file can install the software to any selectable folder, which by default is:
C:\Program Files (x86)\BmCom\

Program settings are shared between all local users, and the destination will depend on the current operating system, for example:

Windows 7: C:\ProgramData\BmCom\

Windows XP: C:\Documents and Settings\All Users\Application Data\BmCom\

The Setup will grant read and write access to the settings folder for all local users by default.

4 Settings

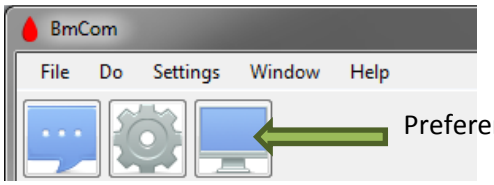
4.1 Communication

The Biomixers can communicate with BmCom through the following interfaces:

Feature	Function	BM330	BM330-1
Data Collection	Collects blood donation results from up to 31 BioMixers.	RS-485/Xbee	RS-485/Xbee/SD-card
Configuration	Configure BioMixer properties, displayed barcode texts and validation of scanned barcodes.	RS-232	USB/ RS-232
Firmware update	Upgrades the BioMixer internal firmware.	N/A	USB

No matter if the PC's built in interface, or an external converter is used, each interface must be detected as a COM-port by the operating system. The OS will enumerate the ports and BmCom relays that the user can select the correct port among all others. The preferred way of finding the port number is to open the Windows device manager, where all ports are listed with additional detailed information.

The port selection in BmCom is available through the Preferences window.



Data collection, configuration and firmware update can use different ports, or share the same one.

Data collection and configuration runs independent from each other in the background. Sharing port will limit this possibility, forcing the user to stop data collection once the configuration window is opened. The same limit states with firmware update, but only while it is active.

It is necessary to connect, and power on all converters including the BioMixer itself, before assigning port; since only the current available ports are selectable in the list. Baud rate must always match the current BioMixer setting (configured in setup-menu). Choosing an incorrect port or baud rate cannot always be automatically detected so It is therefore strongly recommended to verify the complete setup upon any change.

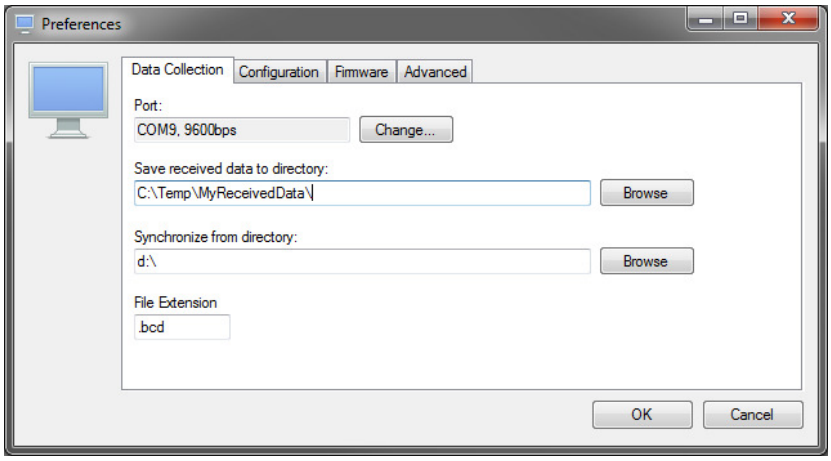
4.2 Compatibility

The old PC-program BM330inf had a feature regarding the data output from collection data. The line ends for each row in the data file was coded as CR CR LF (Carriage Return, Line Feed, Line Feed) instead for the more modern way where a new line is coded as CR LF. When opening the output files in i.e. notepad this is normally not seen, but when other data management systems are importing the output files, there might be some trouble.

Therefore, BMcom can be configured to manage both versions, which is done in Preferences/Additional settings.

5 Data Collection

Blood donation data can be collected through COM-port or SD-card once it is inserted into the computers reader.



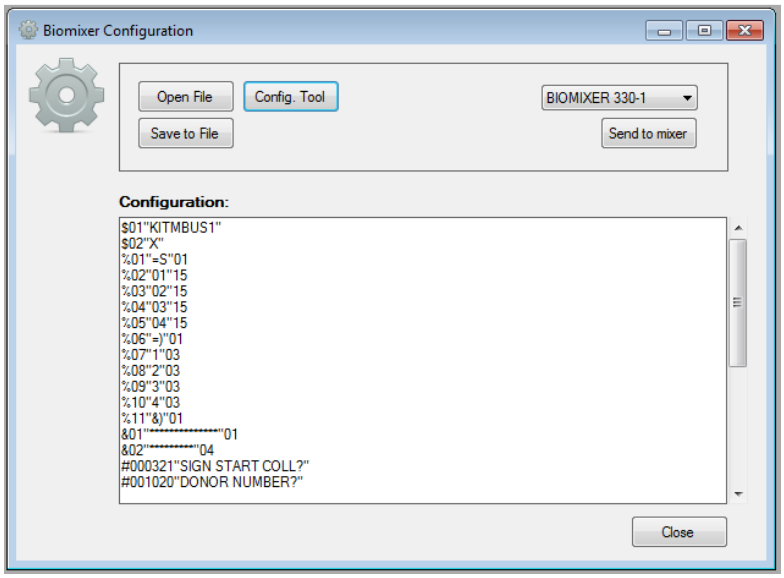
Collected data is stored into the *received data directory*, were each filename is inherited from the Biomixer identity with the selectable file extension. BioMixers without an identity is label as *noname.* New data is always appended to the end of the file using ISO 8859-1 encoding.

Automatic collection from SD-card requires its drive letter to be assigned to the *synchronize directory* in the preferences window.

Last received data coming from COM-port can be monitored by opening the window *View DataCollection*, the raw data is identical to the data appended to the file.

6 Configuration

The BioMixer has a built in set of properties which for example identifies the unit among others. It also stores a list of configurable barcode questions. Configuration of those is written as a script, either manually in the text-editor or generated by the configuration tool. The complete syntax of the script language is available in Appendix 1.



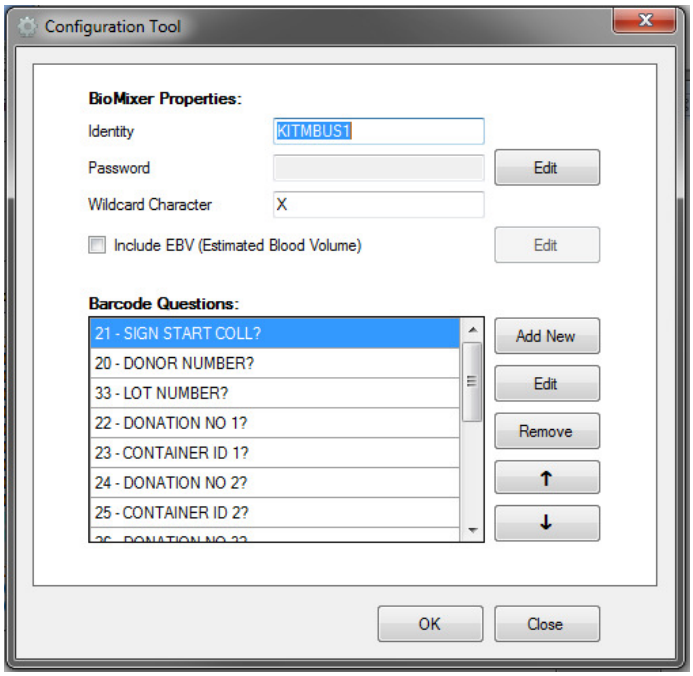
All configurations sent from the BioMixer are displayed automatically in the textbox. Configuration can be loaded and stored to file, edited or written back. It can also be viewed or changed in the configuration tool.

6.1 Configuration Tool

The BioMixer has a set of properties which can be set by the tool:

Property	Function
Identity	Identifies the BioMixer, collected donation data can be tracked to the specific unit.
Password	Prevents the user from entering the BioMixer's setup menu, changing any build-in settings.
Wild Card character	Barcodes containing only this character is accepted as a valid barcode.
Estimated Blood volume (EBV)	Settings for calculating and scanning barcodes for EBV.

The tool also lets you build a list of Barcode questions. New Questions can be added, edited, removed or sorted.



The barcode questions are processed one at a time by the BioMixer: Its text is displayed, and scanned barcodes are validated and added to the donation data before continuing to the next question.

6.2 Barcode questions

Each question is identified with an information code, for example 21. This makes it possible to link the scanned barcode data to the actual question when later opening the collected donation data. Excluding the information code might be useful if the scanning itself contains the information code.

A question can be:

- Used before or after the actual collection.
- Barcode can be scanned as a complete string, or character wise to form a final result finalized by scanning the wildcard character.
- Scanned strings can be validated to be of a fixed or dynamic length.
- The Question can be single: completes after first valid scanning, or repeated: until it expires or the stop text is scanned.

A question can have up to 4 barcode validators, which can be of types:

Validator	Function
Absolute	Barcode must contain an identifier at a specific start index of the string.
Relative	A number of characters at the given start index must match the previous scanned barcode using the same relative validator. This means that the first relative control act as a gauge for all subsequent questions using the same relative validator.

6.3 Estimated blood volume

The BioMixer can estimate the donator's blood volume by using any of the two formulas: Nadler or Stein Holmes. The calculation is based on gender, height and weight, and its inputs needs to be scanned, and thereby requires three additional barcode questions. The configurable sample volume is automatically subtracted from the calculation result.

Two additional parameters need to be set:

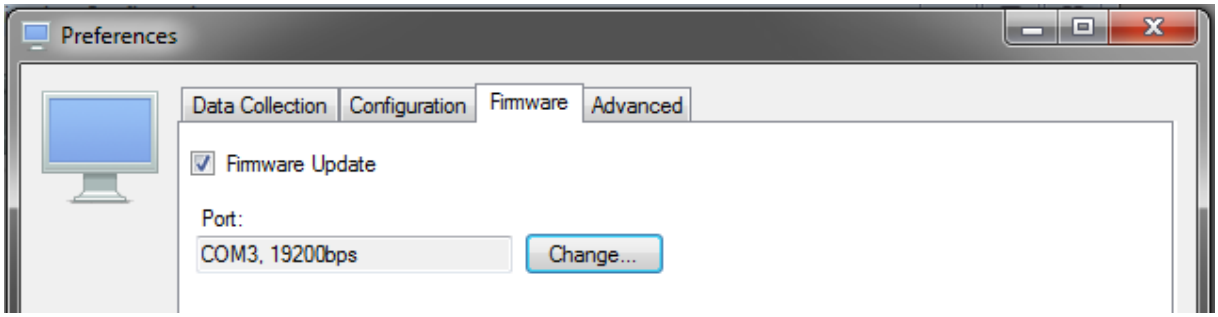
- The collected blood volume which is acceptable to collect is set as the percentage of the estimated volume.
- The percentage of the bag volume that is acceptable to fill.

7 Firmware Update

Note: This feature is only available for BM330-1.

Upgrading the BioMixer's internal firmware shall be handled with care, and avoided unless really needed.

An active choice by the user:



Will make the firmware update accessible:

Warning: All BioMixer settings will be lost when the firmware is updated!

Firmware update procedure:

1. On the BioMixer, hold down START and RESET while the power is switched on. The BioMixer is now waiting for new firmware.
2. Browse for the file. Make sure that it is not compressed. Upload it, by clicking OK.
3. Don't interfere while it is updating, it can take a few minutes.
4. The firmware is now upgraded and the BioMixer needs to be re-configured and calibrated.

8 Appendix 1

8.1 ISBT-128 standard

8.1.1 Some areas of usage

ISBT-128 is a standard for defining barcodes for blood handling.

To be able to carry out some controls according to this standard extended functionality has been added to the configuration file.

The extensions are called 'absolute controls' and 'relative controls'.

These can, according on how they are implemented, be used in a very flexible way also for other applications than ISBT-128. They are however primarily intended for ISBT-128.

- Control of the data identifier for the scanned barcode.
- Control that the donation identification number are the same for all containers in a set.
- Control that the container identity are the same in a set.
- Control of the flag digits in a donation number (00 is treated as a neutral number).
- Control of the flag digits in the container identity.
- Missing container (3 container set) can be scanned as XXXXXXXXXXXXXXXX, (16X) = "No donation identification number" and XXXXXXXXXXXX, (12X) = "No container identity". (X can be set to an optional character).

8.1.2 Data identifiers

According to ISBT-128 the following data identifiers are valid.

=α : Donation Identification Number (Blood ID), where α is the first character
A-Z or 0-9 for the country code or the blood collection organisation.
=% : Blood Groups.
=> : Expiration date.
&> : Expiration date and time
=* : Donation Collection date.
&* : Donation Collection date and time.
=< : Product Code
=) : Manufacturer's Identity and Container Information.
&) : Lot Number
=& : Concatenation programming bar code

8.1.3 Donation Identification Number

Donation identification number data structure is =α pppp yy nnnnnn ff K.

=α	:	Data identifier
pppp	:	Country/collection facility identification together with α
yy	:	Year
nnnnnn	:	6 digit unit serial number
ff	:	Flag digits identifying the container 01-04 in a set. Can also identify extra tests if 07 (test tube). 00 indicates that flag digits are not used.
K	:	Is a check character only to be used for manual entry check. It is not included in the barcode.

8.1.4 Manufacturer's Identity and Container Information

Manufacturer's identity and container information data structure is =) b qq wwwwwww.

=)	:	Data identifier
b	:	Container number in the set
qq	:	Manufacturers identity
wwwwwww	:	Manufacturers catalog number

8.1.5 Lot number

Lot (batch) -number data structure data structure is &) xxxxxxxxxx.

&)	:	Data identifier.
xxxxxxxxxx	:	Manufacturers lot number, 10 characters.

8.2 Configuration

8.2.1 Example #1 (without ISBT-128 extensions)

$01"KSlab034"	Identity set to KSlab034.
$03"1234"	Password set to 1="NEW", 2="SEND", 3="SET", 4="START".
#001020"DONOR NUMBER?"	0=Before collection, 0=Single question, 10=No. of characters, 20=Information code.
#000321"SIGNATURE 1?"	0=Before collection, 0=Single question, 03=No. of characters, 21=Information code.
#001022"COLLECTION NO.?"	0=Before collection, 0=Single question, 10=No. of characters, 22=Information code.
#000323"COMPONENT CODE?"	0=Before collection, 0=Single question, 03=No. of characters, 23=Information code.
#000924"BATCHNO. BLOODBAG?"	0=Before collection, 0=Single question, 09=No. of characters, 24=Information code.
#139900"EXTRA TESTS?"!NO!	1=After collection, 3=Rep. question, 99=1-17 characters, 00=No inf. code, NO=Stop text.
#100131"COLL. RESULT?"	1=After collection, 0=Single question, 01=No. of characters, 31= Information code.
#100332"SIGNATURE 2"	1=After collection, 0=Single question, 03=No. of characters, 32= Information code.

Explanation:

Line 1: String for programming identity:

$01"*iiiiiiii*" <CRLF>

$01 indicates the identity parameter. **iiiiiiii** is the identity, consisting of

1-8 letters or digits. **<CRLF>** is carriage return and line feed. (ENTER).

Line 2: String for login to SETUP menu:

$03"*PPPPPPPPPP*" <CRLF>

$03 indicates the password parameter for login to the SETUP menu, consisting of

0-10 digits, which represents buttons. **<CRLF>** is carriage return and line feed. (ENTER).

Enter an empty string to disable the login function ($03"").

P: 1-9 = "NEW", "SEND", "SET", "START", "PAUSE", "CLAMP", "+", "-", "SETUP".

Default setting is 1245, i.e. NEW-SEND-START-PAUSE

Line 3 - 10 (max 18 lines): String for programming barcode questions:

#*ABCCDD*"*EEEEEE..E*" <CRLF> *or* **#*ABCCDD*"*EEEEEE..E*"!*FFFFF*! <CRLF>**

# Indicates programming barcode questions.

A: Question before or after collection, 0 is before and 1 after.

B: Single or repeated question. 0 = single, 1-9 = repeated. If single question is selected then the sequence continue to next question after barcode scanning, and if it is a repeated question it is terminated when scanning the stop text or the assigned number of 1-9 repeats expires.

The stop text is only supplied for repeated questions, at the end of the programming string, and surrounded by !!.

A maximum of two repeated questions are allowed, one before collection and one after.

CC: The expected number of characters the scanning shall contain. If data does not correspond then an error is indicated with 3 short beeps and *ERR* is viewed on the BM330 display. Maximum 17 characters can be used. **99** indicates that the scanning can have a dynamic length of 1-17 characters. CC is always sent as two digits when programming.

DD: Information code can be chosen between **20-99**. This code is used to mark barcode

scannings for identification when it is processed later on.

If the information code is set to **00** it is excluded from the information. This might be useful if the scanning itself contains the information code, consisting of two digits followed by a space.

Information code **01-19** is reserved for BM330s own data according to the following scheme:

Information code	***Argument***	**Function**
01	*iiiiiiii*	Identity.
02	*DD/MM YYYY hh:mm*	Day, month, year, hour and minute for start of collection.
03	*mm:ss*	Minute and second is time of collection from start to stop.
04	*ppp*	Preset volume in ml.
05	*vvv*	Collected volume in ml.
06	*fff*	Mean flow during collection in ml/minute.
07	flowfail[1]	Flowfail during collection.
08	timefail[2]	Timefail during collection.
09	collection interrupted[3]	Collection interrupted.
10	*DD/MM YYYY hh:mm*[4]	Calibration time.
11	*DD/MM YYYY hh:mm*	Collection stop time.
12	aaa	Tare Volume in ml.
13	Second stick[5]	Second stick.
14-18		Reserved.
19	Checksum error[6]	Checksum error.

Table 1: Information Codes

"EEEE..E": Question text to display. Maximum 20 characters and surrounded by "".

!FFFFF!: Stop text used if it is a repeated question. Maximum allowed length is 5 characters, surrounded by !!. If a repeated question is asked both before and after collection they have each one their own stop text. The stop text is not included for a single question.

<CRLF>: Each line is ended with carriage return and line feed (ENTER).

[1] Or 0 if ok.
[2] Or 0 if ok.
[3] Or 0 if ok.
[4] Or --- if not calibrated.
[5] Or 0 if not second stick.
[6] Or 0 if ok.

8.2.2 Example #2 (with ISBT-128 extensions)

$01"KITMBUS1"	*See example #1.*
$02"X"	Wildcard character for controls.
$03"1234"	*See example #1.*
%01"=S"01	Absolute control, data identifier must be =S.
%02"01"15	Absolute control, number must end with 01.
%03"02"15	Absolute control, number must end with 02.
%04"03"15	Absolute control, number must end with 03.
%05"04"15	Absolute control, number must end with 04.
%06"=)"01	Absolute control, data identifier must be =).
%07"1"03	Absolute control, third digit must be 1.
%08"2"03	Absolute control, third digit must be 2.
%09"3"03	Absolute control, third digit must be 3.
%10"4"03	Absolute control, third digit must be 4.
%11"&)"01	Absolute control, data identifier must be &).
&01"**********"01	Relative control, 14 characters starting at position 01 must be equal.
&02"*********"04	Relative control, 9 characters starting at position 04 must be equal.
#000321"SIGN START COLL?"	*See example #1.*
#001020"DONOR NUMBER?"	*See example #1.*
#001233"LOT NUMBER?";%11	Control 12 characters and data identifier.
#001622"DONATION NO 1?";%01,%02,&01	Control 16 characters and data identifier, flag digits and equal characters.
#001223"CONTAINER ID 1?";%06,%07,&02	Control 12 characters and data identifier.
#001624"DONATION NO 2?";%01,%03,&01	Control 16 characters and data identifier, flag digits and equal characters.
#001225"CONTAINER ID 2?";%06,%08,&02	Control 12 characters and data identifier.
#001626"DONATION NO 3?";%01,%04,&01	Control 16 characters and data identifier, flag digits and equal characters.
#001227"CONTAINER ID 3?";%06,%09,&02	Control 12 characters and data identifier.
#001628"DONATION NO 4?";%01,%05,&01	Control 16 characters and data identifier, flag digits and equal characters.
#001229"CONTAINER ID 4?";%06,%10,&02	Control 12 characters and data identifier.
#009936"SYST BLOODPRESSURE?"	*See example #1.*
#009937"DIAST BLOODPRESSURE?"	*See example #1.*
#059900"EXTRA TEST?"!NO/OK!	*See example #1.*
#100332"SIGN STOP COLL?"	*See example #1.*
#109931"DONATION RESULT?"	*See example #1.*

Explanation:

Line 2: String for programming wildcard character in ISBT-128 controls:

```
$02"X"<CRLF>
```

$02 indicates which character (**X**) to be accepted as a 'wildcard' character.

E.g. to define container 4 in a 3 container set.

("No donation identification number" and "No container identity").

If the answer from an ISBT-128 question contains **X** in all scanned positions, then it is interpreted as a correct answer, i.e. the same as if the answer really had contained a correct substring. **<CRLF>** is carriage return and line feed (ENTER).

Line 4 - 14 (max 20 lines): String for programming ISBT-128 absolute controls:

```
%NN"SS..S"PP<CRLF>
```

% indicates programming absolute controls.

Inserted at the beginning of the configuration file, after identity but prior to question

definitions, and defines an absolute control, number **NN** (01-20), which means that the read barcode must contain the substring **SS..S** (1-5 characters) starting at position **PP** (01-17).

This control might be used to verify that the read barcode has the right data identifier and

container number etc.

E.g. %01"=)1"01 means that absolute control no. 01 verifies that container id has the right data identifier =) and is container 1.

Line 15 - 16 (max 20 lines): String for programming ISBT-128 relative controls:

&NN"**..*"PP<CRLF>

& indicates programming relative controls.

Inserted at the beginning of the configuration file, after identity but prior to question

definitions, and defines a relative control, number **NN** (01-20), which means that the first read barcode during a collection with this control acts as a gauge for the following readings having the same control. Relevant characters is specified with * (max 17 characters) and **PP** (01-17) gives the start position of the substring. **<CRLF>** is carriage return and line feed (ENTER).

E.g. &01"**********"01 means that relative control no. 01 verifies that the 14 first characters is the same as the ones given by the first answer where this control was defined, i.e. to verify that all containers in a set has the same donation number.

Line 17 - 32 (max 20 lines): String for programming barcode questions and activating ISBT-128 controls:

..;%NN,..<CRLF> or ***..;&NN,..<CRLF>***

After the ordinary programming of barcode questions (see example #1) the usage of controls defined earlier in the configuration file can be activated (after **;**).

%NN and **&NN** specifies type and number of the activated control (max 4 per line).

If several controls is activated in one line, then they are separated by **,**.

<CRLF> is carriage return and line feed (ENTER).

8.2.3 Example #3 (EBV)

Note: EBV is only available on BM330-1.

$01"KSlab034"	*See Example #1 (without ISBT-128 extensions)*
$02"X"	*See Example #2 (with ISBT-128 extensions)*
$03"1234"	*See Example #1 (without ISBT-128 extensions)*
$04[21"MALE,FEMALE",22L,23I,NAD,30,15,9]	Control string for EBV
#009921"DONOR GENDER?"	Data container for EBV
#209922"DONOR WEIGHT?"	Data container for EBV
#209923"DONOR HEIGHT?"	Data container for EBV

Explanation:

Line 4: String for EBV-configuration

$04[AA"M-text,F-text",BBw,CCh,SS,DD,EE,FF]<CRLF>

$04: Indicates the identity for EBV (Estimated Blood Volume)

AA: Information code for donor gender. **AA** must be defined as a barcode question

M-text: The barcode for donor gender male

F-text: The barcode for donor gender female

BB: Information code for donor weight. **BB** must be defined as a barcode question

w: Unit used for weight. Allowed values are L for lbs and K for kg

CC: Information code for donor height. **CC** must be defined as a barcode question

h: Unit used for height. Allowed values are I for inch and C for cm.

SS: String representing which formula to use. Allowed values are NAD for Nadler and HOL for Stein Holmes

DD: Sample volume in ml, always two digits. This volume is subtracted from the total blood volume

EE: Percentage of estimated blood volume that is ok to collect, always two digits

FF: Percentage of bag volume that is accepted to fill. If e.g. FF is 9 and the set volume is 450, then it is ok to collect from 410ml to 491ml

<CRLF> is carriage return and line feed (ENTER).

Line 5 -7 (max 18 lines): String for programming barcode questions:

#ABCCDD"EEEEEE..E"<CRLF>

#: Indicates programming barcode questions.

A: Point of time for question: 0=Before collection, 1=After collection, 2=Before collection, character wise, 3=After collection, character wise. Character wise means that character for character is

scanned to form a final result. A scanning sequence is confirmed with the wildcard character defined by $02. Ex: to scan 123, four barcodes are scanned: one for 1, one for 2, one for 3 and one to confirm, e.g. X. If **A** is 2 or 3, **B** must be a single question.

B: Single or repeated question. 0 = single, 1-9 = repeated. If single question is selected then the sequence continue to next question after barcode scanning, and if it is a repeated question it is terminated when scanning the stop text or the assigned number of 1-9 repeats expires.

The stop text is only supplied for repeated questions, at the end of the programming string, and surrounded by !!.

A maximum of two repeated questions are allowed, one before collection and one after.

If **A** is 2 or 3, **B** must be a single question

CC: The expected number of characters the scanning shall contain. If data does not correspond then an error is indicated with 3 short beeps and *ERR* is viewed on the BM330 display. Maximum 17 characters can be used. **99** indicates that the scanning can have a dynamic length of 1-17 characters. **CC** is always sent as two digits when programming.

DD: Information code can be chosen between **20-99**. This code is used to mark barcode

scannings for identification when it is processed later on.

If the information code is set to **00** it is excluded from the information. This might be useful if the scanning itself contains the information code, consisting of two digits followed by a space.

<CRLF> is carriage return and line feed (ENTER).

8.3 Analyzing collected data

Each line of the collected data starts with an information code. 0-19 is reserved according to information code table at page 14. Information codes ≥20 corresponds to the scanned barcode questions, and they are highlighted with **bold text** in this example. (Note that this is only fabricated test values, not values from a real collection).

Please note that all mixer does not produce all fields, it is e.g. only BM330-1 that will have the fields 10-12 and 19.

01 KSlab034	Identity is KSlab034.
02 18/2 2011 14:32	02=Information code, 18/2 2011 14:32=Date and time.
03 09:34	03=Information code, 09:34=Collection time.
04 450	04=Information code, 450=Preset volume in ml.
05 450	05=Information code, 450=Collected volume in ml.
06 47	06=Information code, 47=Mean flow in ml/min.
07 0	07=Information code, flowfail (0 if ok).
08 timefail	08=Information code, timefail (0 if ok).
09 0	09=Information code, collection interrupted (0 if ok).
10 16/2 2011 08:27	10=Information code, 16/2 2011 08:27=Calibration time
11 18/2 2011 14:41	11=Information code, 18/2 2011 14:41=Collection stop time
12 97	12=Information code, tare volume in ml
13 Second stick	13=Information code, indicates second stick.
19 0	19=Information code, checksum error (0 if ok)
20 **5803030200**	20=User def. information code, 5803030200=Donor no. (from barcode reader).
21 **KBS**	21=User def. information code, KBS=Signature before collection (from barcode reader).
22 **0000000012**	22=User def. information code, 0000000012=Collection number (from barcode reader).
23 **101**	23=User def. information code, 101=Component code (from barcode reader).
24 **8895K11KK**	24=User def. information code, 8895K11KK=Bloodbag batch no. (from barcode reader).
40 anti-HTLV	40 anti-HTLV=Extra test (information code replaced by text from barcode reader).
41 anti-HBC	41 anti-HBC=Extra test (information code replaced by text from barcode reader).
30 **0**	30=User defined information code, 0=Collection result (from barcode reader).
31 **KBS**	31=User defined information code, KBS=Signature after collection (from barcode reader).

Explanation:

<CRLF> is used here to indicate carriage return and line feed.

Line 1: Identity:

```
01 iiiiiiii<CRLF>
```

iiiiiiii: Identity, 1-8 characters.

Line 2: Date and time for start of collection:

```
02 DD/MM YYYY hh:mm<CRLF>
```

DD: Day, 1-2 digits.

MM: Month, 1-2 digits.

YYYY: Year, 4 digits.

hh: Hour, 2 digits.

mm: Minute, 2 digits.

Line 3: Collection time from start to stop:

```
03 mm:ss<CRLF>
```

mm: Minute, 1-2 digits.

ss: Second, 2 digits.

Line 4: Preset volume:

```
04 ppp<CRLF>
```

ppp: ml, 1-3 digits.

Line 5: Collected volume:

```
05 vvv<CRLF>
```

vvv: ml, 1-3 digits.

Line 6: Mean flow during collection:

```
06 fff<CRLF>
```

fff: ml/minute, 1-3 digits.

Line 7: Flow fail during collection:

```
07 Flowfail or 0<CRLF>
```

Line 8: Time fail during collection:

```
08 Timefail or 0<CRLF>
```

Line 9: Collection interrupted:

```
09 Collection interrupted or 0<CRLF>
```

Line 10: Calibration time:

```
10 DD/MM YYYY hh:mm or ---<CRLF>
```

DD: Day, 1-2 digits.

MM: Month, 1-2 digits.

YYYY: Year, 4 digits.

hh: Hour, 2 digits.

mm: Minute, 2 digits.

---: If function calibration lockout is disabled

Line 11: Date and time for stop of collection:

```
11 DD/MM YYYY hh:mm <CRLF>
```

DD: Day, 1-2 digits.

MM: Month, 1-2 digits.

YYYY: Year, 4 digits.

hh: Hour, 2 digits.

mm: Minute, 2 digits.

Line 12: Tare volume:

12 ttt *or* ---***<CRLF>***

ttt: ml, 1-3 digits.

---: If function store tare weight is disabled

Line 19: Checksum of stored collection:

19 Checksum error *or* **0*<CRLF>***

Line 20 - nn: Configurable information:

cc *or nothing* ***iiiiiiiiiiiiiiii<CRLF>***

cc: Information code, 2 digits = 20-99.

iiiiiiiiiiiiiiii: 1-17 characters (from barcode reader).